

## Introdução à Ciência da Computação I (Prática)

### *LISTA DE EXERCÍCIOS - Nro. 04*

#### **[Rotinas: Passagem de Parâmetros]**

##### **[Passagem de Parâmetros por Valor]**

1. Faça um programa com uma sub-rotina (função) que receba 3 valores de entrada e retorne o maior valor entre estes três valores.
2. Faça um programa com uma sub-rotina (função) que calcule X elevado à Y =  $X^Y$ . Leia 2 valores de X e Y e exiba o resultado da chamada da sub-rotina na tela. Exemplo: 2 elevado à 3 é igual à  $2*2*2 = 8$ .
3. Faça um programa que peça para ler 2 notas e 2 pesos (1 para cada nota) e depois mostre:
  - A média aritmética simples;
  - A média ponderada entre os dois valores (nota 1 com peso 1 e nota 2 com peso 2);
  - O valor necessário para recuperar a pior nota e passar com média igual ou superior a 6.0 considerando o uso da média aritmética simples (notas com pesos iguais);
  - O valor necessário para recuperar a pior nota e passar com média igual ou superior a 6.0, considerando o uso da média ponderada (nota 1 => peso 1, nota 2 => peso 2);
  - Qual das duas médias (aritmética, ponderada ou tanto faz) é mais benéfica para o aluno, ou seja, qual das duas médias deixa este aluno com a melhor nota.Faça este programa de forma modular, ou seja, usando uma sub-rotina (função) para o cálculo de cada tarefa descrita acima.
4. Faça uma sub-rotina “Br\_UpCase” que dado um caracter qualquer retorne o mesmo caracter sempre em maiúsculo, aceitando inclusive os caracteres acentuados da língua portuguesa (por exemplo: á, é, í, ó, ú, ç, ã, õ, â, ê, ô, à, ù). Faça um programa que leia uma palavra (string) e chame esta sub-rotina para cada um dos caracteres desta palavra, exibindo o resultado após a conversão para maiúsculo. Dicas:  
Letra = Palavra[X]; { Copia o caracter índice X da string “Palavra” para a variável tipo char “Letra” }  
Palavra[5] = Letra; { Copia o conteúdo da variável tipo char “Letra” para o quinto caracter da string armazenada na variável “Palavra” }  
Lembre-se: A string armazenada em uma variável pode ser menor que o tamanho reservado para ela.  
O final da string é representado pelo caracter ‘\0’ (fim de string).
5. Transforme todo o programa anterior de conversão de strings com caracteres em minúsculo para strings com caracteres em maiúsculo, em uma função. Esta função recebe uma string de entrada e devolve a string convertida para letras maiúsculas.
6. Faça um programa para o cálculo do fatorial e verifique se este número é primo ou não. Crie uma rotina separada que realize o cálculo do fatorial de um número e uma outra rotina separada que indique se ele é ou não um número primo. Verifique se o resultado do fatorial é primo.

7. Escrever um programa que obtenha a data atual (ler dia, mês e ano) e exiba ela na tela no formato textual por extenso. A escrita da data por extenso deve ser realizada por um procedimento separado. Exemplo: Data: **01/01/2000** => Imprimir: **Sábado, 1 de janeiro de 2000**.
8. Modificar o programa anterior para que seja lida a data de hoje, o nome de uma pessoa, e a data de seu nascimento. Em seguida o programa deve produzir uma mensagem como a que segue logo abaixo.

Nome: *Fulano da Silva*  
Data de nascimento: *01 01 2008*  
Data de hoje: *16 09 2008*

O **Fulano da Silva**, nascido na **Terça-Feira, 01 de janeiro de 2008** foi registrado neste cartório na data de hoje, **Terça-Feira, 16 de setembro de 2008**.

9. Faça um programa que verifique uma determinada data fornecida pelo usuário, determinando se ela é válida ou não. Validar o dia (1 à 31, conforme o mês), mês (1 à 12) e ano (de 1 ao ano corrente), verificando se esta data é uma data correta. Lembre-se de verificar se o ano é bissexto ao validar a data.
10. Usando o programa que você fez para a validação da data (exercício 9), transforme este programa em uma função que retorne um valor indicando se a data é válida ou não. Altere o programa anterior (exercício 8) de maneira que este valide a data de nascimento fornecida pelo usuário. Se a data for inválida, leia novamente a data de nascimento, valide e repita o processo tantas vezes quantas forem necessárias, até que o usuário forneça uma data correta. Faça o mesmo com a data do dia de hoje fornecida pelo usuário.

### [Passagem de Parâmetros por Referência]

11. Faça um programa que leia 2 valores e chame uma sub-rotina que receba estas 2 variáveis e troque o seu conteúdo, ou seja, esta rotina é chamada passando duas variáveis A e B por exemplo, e após a execução da rotina A conterá o valor de B e B terá o valor de A.
12. Altere o programa anterior de forma a ler 4 valores em 4 variáveis: A, B, C, D. Use a rotina “troca\_valores” implementada no programa anterior para trocar os valores de A com B, depois de C com D, mostre como ficaram os valores das variáveis, e para concluir destrua seus conteúdos mostrando novamente na tela como ficaram estas variáveis.

Valor1: **10**  
Valor2: **20**  
Valor3: **30**  
Valor4: **40**  
Trocando Valor1 com Valor2 e Valor3 com Valor4. Resultado:  
Valor1: **20** - Valor2: **10** - Valor3: **40** - Valor4: **30**  
Trocando Valor1 com Valor2 e Valor3 com Valor4. Resultado:  
Valor1: **10** - Valor2: **20** - Valor3: **30** - Valor4: **40**

13. Faça um programa que leia dois valores e chame uma sub-rotina que receba estes 2 valores de entrada e retorne o maior valor na primeira variável e o segundo maior valor na segunda variável. Escreva o conteúdo das 2 variáveis na tela.
14. Faça um programa que leia três valores e chame uma sub-rotina que receba estes 3 valores de entrada e retorne o maior valor entre estes três valores na primeira variável, o segundo maior valor na segunda variável e o terceiro maior valor na terceira variável. Exibir os valores ordenados na tela (sugestão: use a sub-rotina troca\_valores).

15. Faça um programa que possua uma rotina para ler 2 notas (chamar uma rotina dedicada a ler 2 notas válidas e que devolva os 2 números lidos), uma outra rotina que dada estas 2 notas vá calcular a média simples e a média ponderada onde a segunda nota tem peso 2  $\Rightarrow (n1 + n2 * 2) / 3$ , e uma última rotina que vá exibir na tela todos os dados que foram fornecidos e calculados: rotinas `le_notas`, `calc_media` e `exibe_dados`.
16. Faça um programa que leia 5 números. Este programa deve ter um menu que permita ao usuário escolher qual opção de cálculo que ele deseja realizar: média aritmética simples, média ponderada (ler os pesos associados a cada nota que serão informados pelo usuário), desvio padrão, maior valor e menor valor. A leitura dos 5 valores também deve ser uma das opções do menu. Exemplo:

```
>> Estatística <<
1 – Entrar com os dados (5 valores)
2 – Calcular a média aritmética simples dos dados
3 – Calcular a média ponderada dos dados (fornecer 5 pesos)
4 – Calcular o desvio padrão dos dados
5 – Achar o maior valor
6 – Achar o menor valor
7 – Sair do programa (Fim)
Entre com a sua opção: 1
Valor1: 5.3 Valor2: 8.2 Valor3: 7.3 Valor4: 3.7 Valor5: 7.1
Entre com a sua opção: 2
Média Aritmética Simples: 6.32
Entre com a sua opção: 7
FIM!
```

Obs: Desvio Padrão = Raiz quadrada da divisão do somatório do quadrado das diferenças entre valor e a média de todos os valores, dividido pelo número total de dados. Ver [Wikipedia.org](http://Wikipedia.org)

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2},$$

17. Faça um programa que leia um vetor de 10 números inteiros e ordene este vetor em ordem crescente. Para ordenar o vetor, utilize uma rotina que receba dois números e retorne o menor e o maior. Esta rotina deve ser declarada da seguinte forma: `compara(N1, N2, Menor, Maior)`. Usando esta rotina, compare de dois a dois os números do vetor, até que o maior de todos tenha sido armazenado no final do vetor. Repita esta operação até que o segundo maior tenha sido armazenado na penúltima casa do vetor, depois o terceiro na ante-penúltima, e assim por diante até que todos os dados do vetor estejam ordenados. Este método é conhecido como o método da ordenação da bolha, ou, “Bubble Sort”.
18. Faça um jogo no qual o computador desafia dois usuários para ver quem tem mais memória. O jogo deve gerar um número de 0 à 9 e apresentar ao jogador 1 durante 5 segundos. Depois deve repetir este procedimento para o jogador 2. Em seguida, ele deve pedir ao jogador 1 para informar o número escolhido, e após pedir ao jogador 2 para fazer a mesma coisa. Na segunda rodada, o computador deve gerar mais um número entre 0 e 9, que será “concatenado” à direita do primeiro número. Então repetiremos o procedimento de exibição e teste de memorização para os dois usuários. O programa deve continuar adicionando números (casas adicionadas ao final do número) até que um dos dois jogadores não consiga mais memorizar perfeitamente o número. Para concluir, o programa deve informar qual dos dois jogadores ganhou, quantas casas foram memorizadas ao total, e em que casa foi que o jogador perdedor errou (primeira casa errada começando da esquerda para a direita). Dicas: procure desenvolver o programa em módulos; armazene a seqüência de números dentro de uma string.  $\Rightarrow$  Seu programa ficou bem modular? Seria fácil de adaptá-lo para 3 ou mais jogadores?

## TESTE DE MESA:

19. Procure *prever o comportamento do programa* abaixo e depois teste o programa na prática. Indique qual é o valor das variáveis A, B e X, Y que é escrito na tela no final da execução do programa.

A: \_\_\_\_      B: \_\_\_\_  
X: \_\_\_\_      Y: \_\_\_\_

---

---

Programa:

```
#include <stdio.h>
#include <conio.h>

double x, y;

troca_ab(a,b)
int a,b;
{
    int temp;

    temp=a;
    a=b;
    b=temp;
}

main()
{
    int a,b;

    a=10;
    b=20;

    troca_ab(b,a);

    if ((a > 5) && (b < 15))
    {
        x = a / 3 + b / 2;
        y = a;
    }
    else
    {
        x = a;
        y = (a > b);
    }

    printf("A=%d - B=%d\n", a,b);
    printf("X=%.2lf - Y=%.2lf\n", x,y);

    getch();
}
```